A translation of interaction relationships into SMV language

Zsolt Borsi

In this paper a translation of a particular scenario-based model into SMV language will be presented. SMV is the input language of the NuSMV model-checker tool[4]. Model checkers in general provide a verification way to prove that a given system meets its specification. By using model checking, errors of the system can be detected even in the very early phases of the software development process.

The Unified Modeling Language (UML) provides diagrams to describe the same system from different aspects. The notion of Interaction Overview Diagram (IOD) was introduced in the second version of UML for specifying the relationships between UML interaction diagrams and the control flow passing between them. The notion of IOD is based on activity diagrams. An activity diagram is a directed graph, consisting of nodes and edges. Each node in an IOD is a reference to an interaction and the edges between activity nodes allow the definition of relationships between interactions. This paper does not want to take into account all the constructs included in IODs. On the other hand, some authors (including Whittle in [1]) found useful some additional construct, which are not part of the IODs. The construct included in this extended version of Interaction Overview Diagram (namely EIOD) will be considered in this paper.

The algorythm presented in this paper will translate a hierarchical construct containing EIODs into SMV. In the given construct there is an EIOD at the top level and each node of the top-level EIOD is refined at the underneath level by an EIOD. The algorythm is inspired by two previous works[2, 3]. The first one gives a translation of state charts into SMV. The notation of state charts are similar to IODs in the sense that any state in a statechart may contain whole statecharts like the nodes of EIODs are EIODs. The second one presents an algorythm for translating activity diagrams into SMV language.

The module concept of SMV provides the means for representing every EIOD in a separete module. SMV moduls operate parallel to each other. The top level diagram will be represented by the main module in SMV. The model specification in SMV consists of three parts. First, the possible values of variables should be given to determine the space of states. Then the initial values of the variables and the transition relation should be defined. The state space of our model consists of boolean variables for each node of the top-level EIOD indicating whether the corresponding node is active. All nodes of the respective EIOD are instantiated in that module as well. The specification of these nodes are modelled in separate modules. The transition relation describes the mechanism that passes control along the hierarchical structure.

The idea of using model checking for verification is not new. The novelty of this paper is, that the base of the translation are IODs. Moreover, the transition takes into account additional constructs which are not part of UML, but are used by varios authors.

References

[1] J. Whittle, P.K. Jayaraman. Synthesizing hierarchical state machines from expressive scenario descriptions, *ACM Transactions on Software Engineering and Methodology (TOSEM), v.19 n.3, pp.1-45, 2010.*

[2] E.M. Clark, W. Heinle. Modular Translation of Statecharts to SMV, *Technical Report CMU-CS-00-XXX, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213., 2000.*

[3] R. Eshuis. Symbolic model checking of UML activity diagrams, *ACM Transactions on Software Engineering and Methodology (TOSEM), v.15 n.1, pp1-38, 2006.*

[4] NuSMV Model Checker Home Page, *http://nusmv.fbk.eu*